Attorney Docket No.: 042390.P8098 Patent

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: Carl M. Ellison et al.)
)
Serial No. 09/539,344) Art Unit: 2132
)
Filed: March 31, 2000) Examiner: Grigory Gurshman
)
Title: Managing A Secure Environment)
Using A Hierarchical Executive)
Architecture In Isolated Execution)
Mode)

Mail Stop AF
Commissioner for Patents
P.O Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

IN SUPPORT OF APPELLANT'S APPEAL

TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Sir:

Pursuant to Appellant's Notice of Appeal filed on even date herewith, Appellant hereby submits this Brief in support of its Appeal from the Final Office Action dated September 28, 2004 (hereinafter "the Final Office Action"). Appellant respectfully requests consideration of this Appeal by the Board of Patent Appeals and Interferences and allowance of the claims in the above-captioned patent application.

I. REAL PARTY IN INTEREST

The invention is assigned to Intel Corporation of 2200 Mission College Boulevard, Santa Clara, California 95052, as indicated by the assignment recorded at reel 011106, frame 0545.

II. RELATED APPEALS AND INTERFERENCES

To the best of Appellant's knowledge, there are no appeals or interferences related to the present appeal that will directly affect, be directly affected by, or have a bearing on the Board's decision.

III. STATUS OF CLAIMS

Claims 1-22, 24-40 and 61-99 are pending in the application. Claims 23 and 41-60 were canceled before the Final Office Action. Claims 1-40 and 61-99 have been finally rejected. Claims 1, 21, 61, and 81 are the independent claims. The rejections of all pending claims are appealed.

IV. STATUS OF AMENDMENTS

Two requests for minor claim amendments were recently filed under 37 C.F.R. § 1.116 – the first on December 10, 2004, and the second on December 15, 2004. Appellant has not received any response to indicate whether those amendments have been entered. The amendments involved minor changes to put the claims in better form for consideration in appeal. The changes did not materially affect the substance of the claims, with regard to the grounds of rejection in the Final Office Action. This Appeal Brief therefore anticipates that the requested amendments will be entered.

However, in case the amendments are not entered, the listing of claims provided below includes the markings from the above amendment requests, so that the prior version of the claims will be apparent. Also, all affected claims are prefaced with a parenthetical such as "(previous 1.116 amendment request #1)" or "(previous 1.116 amendment request #2)," as appropriate. In addition, for further reference, Appendix XI below includes a copy of the claims as they existed prior to the changes from the two amendment requests under 37 C.F.R. § 1.116.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Embodiments of the present invention relate to firmware (or other software) that facilitates enhanced security and/or integrity for processing systems. The firmware may execute as part of the process of booting up the processing system, and may provide greater assurance that the software which is loaded at boot time, such as operating system software, has not been tampered with.

For instance, claim 1 pertains to a software module known as a processor executive (PE) (e.g.: FIG 1A, ref. no. 18) that, when executed by a processor (e.g.: FIG. 1C, reference no. 110) in a data processing platform (e.g.: FIG. 1C, ref. no. 100), loads a software module known as an operating system executive (OSE) (e.g.: FIG. 1A, ref. no. 16; FIG. 2, ref. no. 270) in a secure environment of the platform (e.g.: FIG. 1A, ref. no. 15; FIG. 2, ref. no. 250). Claim 1 recites that the OSE serves to manage a subset of an operating system for the platform.

Regarding the secure environment, claim 1 recites that the platform has "a processor capable of selectively operating in one of a normal execution mode and, alternatively, in an isolated execution mode" (e.g.: FIGs. 1A-1C). Claim 1 also recites that the secure environment is associated with an "isolated memory area" that is "accessible to the processor in the isolated execution mode" (e.g.: FIG. 1B, ref. no. 70).

In addition, claim 1 recites a "PE handler" that uses certain components "to verify the PE" (e.g.: FIG. 2, ref. no. 230; FIG. 3, ref. nos. 230). According to claim 1, one of those components is a PE supplement "comprising a PE manifest that represents the PE" (e.g.: FIG. 2, ref. no. 222; FIG. 3, ref. nos. 222, 320, 70).

The dependent claims recite numerous additional details. For instance, claim 4 recites a "PE loader to load the PE into the isolated memory," and "a verifier to verify the PE using the PE manifest." Claim 7 recites that the PE comprises "an OSE loader to load the OSE … into the isolated memory" and "an OSE verifier to verify the OSE."

Additional claims involve various other features for providing enhanced security and/or integrity, such as components for creating or using various keys, components for managing the isolated memory area, etc.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Whether the Examiner erred in rejecting claims 1 and 61 and their dependent claims under 35 U.S.C. § 103(a) as being unpatentable over U.S. patent no. 5,809,546 to Paul Gregory Greenstein et al. (hereinafter "Greenstein") in view of U.S. patent no. 4,419,724 to Michael H. Branigin et al. (hereinafter "Branigin").

B. Whether the Examiner erred in rejecting claim 81 and its dependent claims under 35 U.S.C. § 103(a) as being unpatentable over Greenstein in view of Branigin.

C. Whether the Examiner erred in rejecting claim 21 and its dependent claims under 35 U.S.C. § 103(a) as being unpatentable over Greenstein in view of Branigin.

D. Whether the Examiner erred in rejecting claims 8, 28, 68, 87, and their dependent claims as being unpatentable over Greenstein in view of Branigin.

E. Whether the Examiner erred in rejecting claims 27 and 86 under 35 U.S.C. § 103(a) as being unpatentable over Greenstein in view of Branigin.

F. Whether the Examiner erred in rejecting claims 20, 40, and 80 under 35 U.S.C. § 103(a) as being unpatentable over Greenstein in view of Branigin.

G. Whether the Examiner erred in rejecting claims 10, 30, 70, and 90 and their dependent claims under 35 U.S.C. § 103(a) as being unpatentable over Greenstein in view of Branigin.

H. Whether the Examiner erred in rejecting claims 14, 34, 74, and 94 and their dependent claims under 35 U.S.C. § 103(a) as being unpatentable over Greenstein in view of Branigin.

I. Whether the Examiner erred in rejecting claims 15, 16, 35, 36, 75, 76, 95, and 96 under 35 U.S.C. § 103(a) as being unpatentable over Greenstein in view of Branigin.

J. Whether the Examiner erred in rejecting claims 17, 37, 77, and 97 under 35 U.S.C. § 103(a) as being unpatentable over Greenstein in view of Branigin.

K. Whether the Examiner erred in rejecting claims 4, 24, 64, and 83 under 35 U.S.C. § 103(a) as being unpatentable over Greenstein in view of Branigin.

L. Whether the Examiner erred in rejecting claims 3, 7, 63, and 67 under 35 U.S.C. § 103(a) as being unpatentable over Greenstein in view of Branigin.

M. Whether the Examiner erred in rejecting claims 2, 5-6, 9, 11-13, 18-19, 22, 25-26, 29, 31-33, 38-39, 62, 65-66, 69, 71-73, 78-79, 82, 84-85, 88-89, 91-93, and 98-99 under 35 U.S.C. § 103(a) as being unpatentable over Greenstein in view of Branigin.

N. Whether the Final Office Action establishes a *prima facie* case of obviousness for claims 9, 11-12, 29, 31-32, 69, 71, 72, 82, 84-86, 88-96, and 98.

VII. ARGUMENT

The Claims Are Patentable Over Greenstein and Branigin

The Final Office Action fails to present a *prima facie* case of obviousness for Appellant's claims. Even if Greenstein and Branigin were to be combined, the combination would not disclose or suggest every aspect of the claimed invention.

A. Claims 1 and 61 stand finally rejected under 35 U.S.C. 103(a) as being unpatentable over Greenstein in view of Branigin. Appellant respectfully requests that these rejections be overturned for at least the following reasons.

(i) Regarding a processor executive "executable on a processor" to load an OSE in a secure environment of a platform

As indicated above, the present invention relates to firmware or other software that facilitates enhanced security and/or integrity for processing systems. In particular, Claim 1 recites a processor executive "*executable on a processor*" to load an OSE in a secure environment of a platform, the OSE to manage a subset of an OS running on the platform.

Greenstein pertains to a method for managing input/output (I/O) buffers in shared storage. Branigin pertains to a main bus interface package. Neither Greenstein nor Branigin discloses or suggests a processor executive that, when executed on a processor, loads an operating system executive in a secure environment of a platform.

The Final Office Action asserts that a processor executive is taught by central processing unit (CPU) 101 in Fig. 1 of Greenstein. However, CPU 101 is merely a hardware component within computer system 100 – namely, a central processing unit. CPU 101 could be used to execute software, but CPU 101 is not itself software. Consequently, CPU 101 could not possibly constitute a processor executive "*executable on a processor*" to load an OSE.

CPUs may also be referred to as "processors." However, a processor simply is not the same thing as a "processor executive" as recited in claim 1. To assert that CPU 101 constitutes a "processor executive" is to assert that a processor can be executable on processor. However, those of ordinary skill in the art plainly understand that processors do not execute processors – processors execute instructions. The Final Office Action nevertheless asserts that a processor executive is the same thing as a processor.

All of the rejections in the Final Office Action therefore seem to be based on the premise that, if a term contains two words, one may disregard the second word without altering the meaning of the term. However, by that same premise, there would be no difference between a "computer technician" and a "computer," or between a "mountain lion" and "mountain." The premise must therefore be faulty, and the rejections invalid. A processor executive is simply not the same thing as a processor.

For instance, claim 1 recites that the processor executive is "executable on a processor to load an operating system executive (OSE) in a secure environment" of a platform. CPU 101 of Greenstein is not "executable on a processor" to do anything, let alone load an OSE in a secure environment of a platform.

(ii) Regarding a PE handler to "verify" the processor executive using a "PE supplement"

In addition, as indicated above, claim 1 recites a "PE handler" that uses certain components "to verify the PE." According to claim 1, those components include a "PE supplement" which comprises "a PE manifest that represents the PE." Thus, claim 1 involves verification of the processor executive software, through use of a PE supplement including a PE manifest. Even if Greenstein and Branigin were to be combined, the combination would not disclose or suggest the use of a PE supplement with a PE manifest to verify a software component such as a processor executive.

The Final Office Action asserts that CPU 101, storage controller 105, and CPU key 309 in Fig. 3. of Greenstein pertain to a PE handler. However, Greenstein says nothing about processor executives, or PE handlers.

The Final Office Action also asserts that a "destination ID," as used in column 18, lines 43-63 of Branigin, somehow pertains to a PE manifest that represents a processor executive. However, Branigin says nothing about processor executives or PE manifests that represent processor executives. Instead, the destination IDs in Branigin are simply used to direct communications to an appropriate hardware destination, in a system with one or more CPUs and a "plurality of Main Storage Units" (MSUs). The destination IDs in Branigin thus have nothing to do with a manifest that represents a software entity in general, let alone a PE manifest that represents a processor executive.

In addition, the Final Office Action asserts that combining the CPU 101, storage controller 105, and CPU key 309 of Greenstein with the destination ID of Branigin would produce a PE handler that verifies a processor executive through use of a PE supplement including a PE manifest. That assertion is not well founded. Combining Greenstein and Branigin would merely produce a system that manages I/O buffers in shared storage, as per Greenstein, while also providing for communications between one or more CPUs and multiple MSUs, as per Branigin. The combination would not disclose or suggest a "PE handler to verify" a processor executive using a "PE supplement," as recited in claim 1.

The Final Office Action therefore fails to establish a *prima facie* case of obviousness for claim 1.

Like claim 1, claim 61 involves a processor executive that executes on a processor and loads an OSE in a platform. Like claim 1, claim 61 also involves verification of the processor executive, through use of a PE supplement with a PE manifest that represents the processor executive.

The Final Office Action rejects claim 61 on the same bases as claim 1. As indicated above, those bases are incorrect. The Final Office Action therefore fails to establish a *prima facie* case of obviousness for claim 61.

For at least the foregoing reasons, the rejections of claims 1 and 61 are improper.

B. <u>Claim 81</u> stands finally rejected under 35 U.S.C. 103(a) as being unpatentable over Greenstein in view of Branigin. Appellant respectfully requests that this rejection be overturned for at least the following reasons.

Like claim 1, claim 81 involves a processor executive that executes on a processor and loads an OSE in a platform. Like claim 1, claim 81 also involves verification of the processor executive, through use of a PE supplement with a PE manifest that represents the processor executive.

In addition, claim 81 clearly recites that it is software (i.e., (i.e., by "instructions" encoded in a "machine accessible medium.") which causes the platform to (a) load the OSE into an isolated memory area of the platform and (b) verify the processor executive.

The Final Office Action does not address these specific features of claim 81. For instance, the Final Office Action does not explain what portion of Greenstein or Branigin allegedly discloses "instructions" which, when executed, cause the platform to load an OSE into an isolated memory area of the platform. The Final Office Action also does not explain what portion of Greenstein or Branigin allegedly discloses "instructions" which, when executed, cause the platform to verify the processor executive.

Instead, the Final Office Action simply lumps claim 81 in with the rejection of claim 1. As indicated above, those bases of that rejection are incorrect. Neither Greenstein nor Branigin discloses or suggests instructions that, when executed, cause a platform to load an OSE. Likewise, neither Greenstein nor Branigin discloses or suggests instructions that, when executed, cause a platform to verify a processor executive.

The Final Office Action therefore fails to establish a *prima facie* case of obviousness for claim 81.

For at least the foregoing reasons, the rejection of claim 81 is improper.

C. Claim 21 stands finally rejected under 35 U.S.C. 103(a) as being unpatentable over Greenstein in view of Branigin. Appellant respectfully requests that this rejection be overturned for at least the following reasons.

Like claim 1, claim 21 involves a processor executive that executes on a processor and loads an OSE in a platform. Like claim 1, claim 21 also involves verification of the processor executive, through use of a PE supplement with a PE manifest that represents the processor executive.

The Final Office Action rejects claim 21 on the same bases as claim 1. As indicated above, those bases are incorrect.

In addition, claim 21 recites that the OSE is loaded into "an isolated memory area of the platform, ... the isolated memory area being accessible to the processor in the isolated execution mode." The Final Office Action asserts that the above aspects are disclosed in Greenstein at elements 105 and 110 in Fig. 1 and elements 402-406 in Fig.

4a. Although the cited elements may in general have something to do with restricting access to storage, they do not disclose loading "an operating system executive (OSE)" into protected storage that is accessible to a processor in "isolated execution mode." Moreover, Greenstein does not disclose loading, into an isolated memory area, a software component (i.e., the OSE) that is "to manage a subset of an operating system (OS) running on the platform."

The Final Office Action therefore fails to establish a *prima facie* case of obviousness for claim 21.

For at least the foregoing reasons, the rejection of claim 21 is improper.

D. Claims 8, 28, 68, and 87 stand finally rejected under 35 U.S.C. 103(a) as being unpatentable over Greenstein in view of Branigin. Appellant respectfully requests that these rejections be overturned for at least the following reasons.

Claims 8, 28, 68, and 87 recite additional details regarding an embodiment of the processor executive. In particular, each of those claims recites specific operations to be performed by the processor executive, such as generating an OSE key, logging an identifier for the OSE in storage, and handling an OSE entry and an OSE exit. In addition, claim 87 clearly identifies the processor executive as software. That is, claim 87 plainly recites that the processor executive is implemented as instructions encoded in a machine accessible medium.

Even if Greenstein and Branigin were to be combined, the combination would not produce the features recited in the independent claims, let alone the combination of those features with the features recited in claims 8, 28, 68, and 87.

The Final Office Action neglects to explain the rejection of claim 87. However, the Final Office Action asserts that Figs. 1, 7, and 8 of Greenstein disclose the features of claims 8, 28, and 68. That assertion is incorrect. As indicated above, Greenstein does not disclose a processor executive that is executable on a processor to load an OSE, as recited in the independent claims. *A priori*, the cited figures to not disclose a processor executive that, when executed on a processor, generates an OSE key, logs an identifier for the OSE in storage, and handles an OSE entry and an OSE exit.

The Final Office Action therefore fails to establish a *prima facie* case of obviousness for any of claims 8, 28, 68, and 87.

For at least the foregoing reasons, the rejections of claims 8, 28, 68, and 87 are improper.

E. Claims 27 and 86 stand finally rejected under 35 U.S.C. 103(a) as being unpatentable over Greenstein in view of Branigin. Appellant respectfully requests that these rejections be overturned for at least the following reasons.

Claim 27 recites additional details regarding an embodiment of the method recited in claim 21 for providing a platform with enhanced integrity. For instance, claim 27 recites the operation of "verifying the OSE after loading the OSE into the isolated memory area." Claim 86 recites similar details regarding an embodiment of the software recited in claim 81.

Even if Greenstein and Branigin were to be combined, the combination would not produce the features recited in the independent claims, let alone the combination of those features with the features recited in claims 27 and 86.

The Final Office Action neglects to explain the rejection of claim 86. However, the Final Office Action asserts that Fig. 21 and column 18, lines 43-63 of Greenstein disclose the features of claim 27. That assertion is incorrect. As indicated above, Greenstein does not disclose a processor executive that is executable on a processor to load an OSE, as recited in the independent claims. Furthermore, Fig. 21 and column 18, lines 43-63 of Greenstein do not disclose "verifying the OSE after loading the OSE into the isolated memory area

The Final Office Action therefore fails to establish a *prima facie* case of obviousness for any of claims 27 and 86.

For at least the foregoing reasons, the rejections of claims 27 and 86 are improper.

F. Claims 20, 40, and 80 stand finally rejected under 35 U.S.C. 103(a) as being unpatentable over Greenstein in view of Branigin. Appellant respectfully requests that these rejections be overturned for at least the following reasons.

Claim 40 recites additional details regarding an embodiment of the method recited in claims 21 and 28 for providing a platform with enhanced integrity. Specifically, claim 40 recites that the storage which is used for logging the OSE identifier is "an input/output controller hub (ICH) external to the processor." Claims 20 and 80 recites similar details.

Even if Greenstein and Branigin were to be combined, the combination would not produce the features recited in the independent claims, let alone the combination of those features with the features recited in claims 20, 40 and 80.

The Final Office Action explains the rejections of claims 20, 40, and 80 by simply asserting that "it is well known in the art to have a chipset including memory controller hub [sic] and input output controller hub [sic]." The Final Office Action fails to address the specific claim language, i.e., that an OSE identifier is to be logged in an input/output controller hub.

As indicated above, Greenstein does not disclose a processor executive that is executable on a processor to load an OSE, as recited in the independent claims. Furthermore, Greenstein does not disclose the concept logging an OSE identifier in an input/output controller hub (ICH). The Final Office Action therefore fails to establish a *prima facie* case of obviousness for any of claims 20, 40, and 80.

For at least the foregoing reasons, the rejections of claims 20, 40 and 80 are improper.

G. Claims 10, 30, 70, and 90 stand finally rejected under 35 U.S.C. 103(a) as being unpatentable over Greenstein in view of Branigin. Appellant respectfully requests that these rejections be overturned for at least the following reasons.

Claim 30 recites that the OSE performs operations comprising: "loading a module into the isolated memory area; managing paging in the isolated memory area; and interfacing with the OS." Claims 10, 70, and 90 recite similar features.

Even if Greenstein and Branigin were to be combined, the combination would not disclose or suggest the features of the independent claims, together with an OSE that performs the operations of loading a module into an isolated memory area, managing paging in the isolated memory area, and interfacing with the OS. The Final Office

Action therefore fails to establish a *prima facie* case of obviousness for any of claims 10, 30, 70, and 90.

For at least the foregoing reasons, the rejections of claims 10, 30, 70, and 90 are improper.

H. Claims 14, 34, 74, and 94 stand finally rejected under 35 U.S.C. 103(a) as being unpatentable over Greenstein in view of Branigin. Appellant respectfully requests that these rejections be overturned for at least the following reasons.

Claim 34 involves a method with operations comprising "executing an isolated create instruction during a process of booting the platform, the isolated create instruction loading the PE handler into the isolated memory area." Claims 14, 74, and 94 recite similar features.

Even if Greenstein and Branigin were to be combined, the combination would not disclose or suggest the features of the independent claims, together with an isolated create instruction to be executed during a process of booting a platform, wherein the isolated create instruction loads a PE handler into an isolated memory area. The Final Office Action therefore fails to establish a *prima facie* case of obviousness for any of claims 14, 34, 74, and 94.

For at least the foregoing reasons, the rejections of claims 14, 34, 74, and 94 are improper.

I. Claims 15, 16, 35, 36, 75, 76, 95, and 96 stand finally rejected under 35 U.S.C. 103(a) as being unpatentable over Greenstein in view of Branigin. Appellant respectfully requests that these rejections be overturned for at least the following reasons.

Claims 35 and 36 refer to the isolated create instruction in terms of an atomic sequence that includes operations such as "reading a thread count register in a chipset to determine if the processor is the first processor in the isolated execution mode; configuring the isolated execution mode; and loading the PE handler into the isolate memory area." Claims 15, 16, 75, 76, 95, and 96 recite similar features.

Even if Greenstein and Branigin were to be combined, the combination would not disclose or suggest the features of the independent claims, together with an atomic sequence including the operations listed above. The Final Office Action therefore fails to establish a *prima facie* case of obviousness for any of claims 15, 16, 35, 36, 75, 76, 95, and 96.

For at least the foregoing reasons, the rejections of claims 15, 16, 35, 36, 75, 76, 95, and 96 are improper.

J. Claims 17, 37, 77, and 97 stand finally rejected under 35 U.S.C. 103(a) as being unpatentable over Greenstein in view of Branigin. Appellant respectfully requests that these rejections be overturned for at least the following reasons.

Claim 37 recites that the atomic sequence also comprises "verifying a loaded PE handler; and transferring control to the loaded PE handler." Claims 17, 77, and 97 recite similar features.

Even if Greenstein and Branigin were to be combined, the combination would not disclose or suggest the features of the independent claims, together with an atomic sequence including the operations of verifying a loaded PE handler and transferring control to the loaded PE handler. The Final Office Action therefore fails to establish a *prima facie* case of obviousness for any of claims 17, 37, 77, and 97.

For at least the foregoing reasons, the rejections of claims 17, 37, 77, and 97 are improper.

K. Claims 4, 24, 64, and 83 stand finally rejected under 35 U.S.C. 103(a) as being unpatentable over Greenstein in view of Branigin. Appellant respectfully requests that these rejections be overturned for at least the following reasons.

Claim 24 recites that the PE handler performs operations such as "loading the [processor executive] into the isolated memory," and "verifying the [processor executive] using the PE manifest." Claims 4, 64, and 83 recite similar features.

Even if Greenstein and Branigin were to be combined, the combination would not disclose or suggest the features of the independent claims, together with a PE handler that performs the operations of loading a processor executive into isolated memory, and verifying the processor executive using a PE manifest. The Final Office Action therefore fails to establish a *prima facie* case of obviousness for any of claims 4, 24, 64, and 83.

For at least the foregoing reasons, the rejections of claims 4, 24, 64, and 83 are improper.

L. Claims 3, 7, 63, and 67 stand finally rejected under 35 U.S.C. 103(a) as being unpatentable over Greenstein in view of Branigin. Appellant respectfully requests that these rejections be overturned for at least the following reasons.

Claim 63 recites an OSE supplement that comprises "an OSE manifest that represents the OSE." In addition, claim 67 recites that the processor executive comprises "an OSE loader to load the OSE and the OSE supplement into the isolated memory area; an OSE manifest verifier to verify the OSE manifest; and an OSE verifier to verify the OSE." Claims 3 and 7 involve similar features, respectively.

Even if Greenstein and Branigin were to be combined, the combination would not disclose or suggest the features of the independent claims, together with an OSE manifest that represents the OSE. The combination also would not disclose or suggest a processor executive that (a) loads an OSE and an OSE supplement into an isolated memory area, (b) verifies the OSE manifest, and (c) verifies the OSE. The Final Office Action therefore fails to establish a *prima facie* case of obviousness for any of claims 3, 7, 63, and 67.

For at least the foregoing reasons, the rejections of claims 3, 7, 63, and 67 are improper.

M. Claims 2, 5-6, 9, 11-13, 18-19, 22, 25-26, 29, 31-33, 38-39, 62, 65-66, 69, 71-73, 78-79, 82, 84-85, 88-89, 91-93, and 98-99 stand finally rejected under 35 U.S.C. 103(a) as being unpatentable over Greenstein in view of Branigin. Appellant respectfully requests that these rejections be overturned for at least the following reasons.

Even if Greenstein and Branigin were to be combined, the combination would not produce the specific features recited in the independent claims, together with the additional features from the claims listed in the preceding paragraph.

The Final Office Action fails to establish a *prima facie* case of obviousness for any of claims 2, 5-6, 9, 11-13, 18-19, 22, 25-26, 29, 31-33, 38-39, 62, 65-66, 69, 71-73, 78-79, 82, 84-85, 88-89, 91-93, and 98-99.

For the foregoing reasons and other reasons, the rejections of claims 2, 5-6, 9, 11-13, 18-19, 22, 25-26, 29, 31-33, 38-39, 62, 65-66, 69, 71-73, 78-79, 82, 84-85, 88-89, 91-93, and 98-99 are improper.

N. Claims 9, 11-12, 29, 31-32, 69, 71, 72, 82, 84-86, 88-96, and 98 stand finally rejected under 35 U.S.C. 103(a) as being unpatentable over Greenstein in view of Branigin. However, the Final Office Action does not provide any explanation for these rejections. In particular, the Final Office Action does not explain which portions of which references allegedly disclose or suggest the numerous different features recited in these claims. The Final Office Action therefore fails to establish a *prima facie* case of obviousness for any of claims 9, 11-12, 29, 31-32, 69, 71, 72, 82, 84-86, 88-96, and 98.

For at least the foregoing reasons, the rejections of claims 9, 11-12, 29, 31-32, 69, 71, 72, 82, 84-86, 88-96, and 98 are improper. Appellant therefore respectfully requests that these rejections be overturned.

Conclusion

Appellant respectfully submits that all pending claims in this patent application are patentable, and requests that the Board of Patent Appeals and Interferences overrule the Examiner and direct allowance of the rejected claims.

If any fee insufficiency or overpayment is found, please charge any insufficiency or credit any overpayment to Deposit Account No. 02-2666.

Respectfully submitted,

Intel Corporation

Date: December 17, 2004

/ Michael R. Barre # 44,023/

Michael R. Barré
Registration No. 44,023
Patent Attorney
Intel Americas, Inc.

Attorney Phone Number: (512) 732-3923

Correspondence Address:

Blakely Sokoloff Taylor & Zafman, LLP
12400 Wilshire Blvd
Seventh Floor
Los Angeles, California 90025-1026

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313 on:

17 DECEMBER 2004
Date of Deposit

Deborah L. Higham
Name of Person Mailing Correspondence

12-17-04
Signature Date

VIII. CLAIMS APPENDIX

1. An apparatus comprising:

a processor executive (PE) executable on a processor to load an operating system executive (OSE) in a secure environment, the secure environment having a fused key (FK) and associated with an isolated memory area in a platform having the processor, the OSE to manage a subset of an operating system (OS) running on the platform, the processor capable of selectively operating in a normal execution mode and, alternatively, in an isolated execution mode, the isolated memory area being accessible to the processor in the isolated execution mode;

a PE supplement comprising a PE manifest that represents the PE; and

a PE handler to verify the PE using the FK and the PE supplement.

2. The apparatus of claim 1 further comprising:

a boot-up code to load the PE handler into the isolated memory area during a process of booting up the platform.

3. The apparatus of claim 1 wherein the secure environment includes an OSE supplement comprising an OSE manifest that represents the OSE.

4. The apparatus of claim 1 wherein the PE handler comprises:

a PE loader to load the PE into the isolated memory area; and

a verifier to verify the PE using the PE manifest.

5. The apparatus of claim 1 wherein the PE handler comprises:

a PE key generator to generate a PE key using the FK;

a PE identifier logger to log a PE identifier in a storage; and

a PE entrance/exit handler to handle a PE entry and a PE exit.

6. The apparatus of claim 5 wherein the PE key generator comprises:

a PE key combiner to combine the PE identifier and the FK, the combined PE identifier and FK corresponding to the PE key.

7. The apparatus of claim 3 wherein the PE comprises:

an OSE loader to load the OSE and the OSE supplement into the isolated memory area;

an OSE manifest verifier to verify the OSE manifest; and

an OSE verifier to verify the OSE.

8. The apparatus of claim 1 wherein the PE comprises:

an OSE key generator to generate an OSE key;

an OSE identifier logger to log an OSE identifier in a storage; and

an OSE entrance/exit handler to handle an OSE entry and an OSE exit.

9. The apparatus of claim 8 wherein the OSE key generator comprises:

a binding key generator to generate a binding key (BK) using a PE key; and

an OSE key combiner to combine the OSE identifier and the BK, the combined OSE identifier and BK corresponding to the OSE key.

10. The apparatus of claim 1 wherein the OSE comprises:

a module loader to load a module into the isolated memory area;

a page manager to manage paging in the isolated memory area; and

an interface handler to handle interfacing with the OS.

11. The apparatus of claim 10 wherein the module comprises one or more modules selected from the group consisting of an application module, an applet module, and a support module.

12. The apparatus of claim 11 wherein the OSE further comprises:

an applet key generator to generate an applet key associated with the applet module.

13. The apparatus of claim 12 wherein the applet key generator comprises:

an applet key combiner to combine an OSE key with an applet identifier identifying the applet module, the combined OSE key and applet identifier corresponding to the applet key.

14. The apparatus of claim 4 wherein the boot-up code comprises:

a PE locator to locate the PE and the PE supplement, the PE locator transferring the PE and the PE supplement into the PE memory at a PE address;

a PE recorder to record the PE address in a parameter block; and

an instruction invoker to execute an isolated create instruction, the isolated create instruction loading the PE handler into the isolated memory area.

15. The apparatus of claim 14 wherein the isolated create instruction performs an atomic sequence, the atomic sequence being non-interruptible.

16. The apparatus of claim 15 wherein the atomic sequence includes operations comprising:

reading a thread count register in a chipset to determine if the processor is the first processor in the isolated execution mode;

configuring the processor in the isolated execution mode; and

loading the PE handler into the isolated memory area.

17. The apparatus of claim 15 wherein the atomic sequence of operations comprises:

verifying a loaded PE handler; and

transferring control to the loaded PE handler.

18. The apparatus of claim 16 wherein the atomic sequence of operations further comprises:

reading a configuration storage in the chipset when the processor is not the first processor in the isolated execution mode; and

configuring the processor according to the configuration storage in the chipset when the processor is not the first processor in the isolated execution mode.

19. The apparatus of claim 18 wherein the chipset includes at least one hub selected from the group consisting of a memory controller hub (MCH) and an input/output controller hub (ICH).

20. The apparatus of claim 8 wherein the storage is in an input/output controller hub (ICH) external to the processor.

21. (previous 1.116 amendment request #2) A method comprising:

loading an operating system executive (OSE) into an isolated memory area of a platform, the platform having a fused key (FK) and a processor capable of selectively operating in a normal execution mode and, alternatively, in an isolated execution mode, the OSE to manage a subset of an operating system (OS) running on the platform, the isolated memory area being accessible to the processor in the isolated execution mode, the loading of the OSE initiated by a processor executive (PE) executing on the processor; and

verifying the PE using the FK and a PE supplement having a PE manifest that represents the PE, the verification to be performed by a PE handler.

22. The method of claim 21 further comprising:

loading the PE into the isolated memory area during a process of booting up the platform.

23. (canceled)

24. The method of claim 21 wherein the PE handler performs operations comprising:

loading the PE into the isolated memory area; and

verifying the PE using the PE manifest.

25. The method of claim 24 wherein the PE handler performs operations comprising:

generating a PE key using the FK;

logging a PE identifier in a storage; and

handling a PE entry and a PE exit.

26. The method of claim 25 wherein generating the PE key comprises:

combining the PE identifier and the FK, the combined PE identifier and FK corresponding to the PE key.

27. (previous 1.116 amendment request #1) The method of claim 21, further comprising:

[[and]]

verifying the OSE after loading the OSE into the isolated memory area.

28. The method of claim 21 wherein the operations performed by the PE comprise:

generating an OSE key;

logging an OSE identifier in a storage; and

handling an OSE entry and an OSE exit.

29. The method of claim 28 wherein generating the OSE key comprises:

generating a binding key (BK) using the PE key; and

combining the OSE identifier and the BK, the combined OSE identifier and BK corresponding to the OSE key.

30. The method of claim 21 wherein the OSE manages the subset of the OS by performing operations comprising:

loading a module into the isolated memory area;

managing paging in the isolated memory area; and

interfacing with the OS.

31. (previous 1.116 amendment request #2) The method of ~~claim 29~~ claim 30 wherein the module comprises one or more modules selected from the group consisting of an application module, an applet module, and a support module.

32. The method of claim 31 wherein the OSE performs further operations comprising:

generating an applet key associated with the applet module.

33. The method of claim 32 wherein:

the OSE combines an OSE key with an applet identifier identifying the applet module, the combined OSE key and applet identifier corresponding to the applet key.

34. The method of claim 21, further comprising:

locating the PE and the PE supplement;

transferring the PE and the PE supplement into the PE memory at a PE address during a process of booting the platform;

recording the PE address in a parameter block; and

executing an isolated create instruction during the process of booting the platform, the isolated create instruction loading the PE handler into the isolated memory area.

35. The method of claim 34 wherein executing the isolated create instruction comprises performing an atomic sequence, the atomic sequence being non-interruptible.

36. The method of claim 35 wherein performing the atomic sequence comprises:

reading a thread count register in a chipset to determine if the processor is the first processor in the isolated execution mode;

configuring the processor in the isolated execution mode; and

loading the PE handler into the isolated memory area.

37. The method of claim 35 wherein performing the atomic sequence comprises:

verifying a loaded PE handler; and

transferring control to the loaded PE handler.

38. The method of claim 36 wherein configuring the processor in the isolated execution mode comprises:

reading a configuration storage in the chipset when the processor is not the first processor in the isolated execution mode; and

configuring the processor according to the configuration storage in the chipset when the processor is not the first processor in the isolated execution mode.

39. The method of claim 38 wherein the chipset includes at least one hub selected from the group consisting of a memory controller hub (MCH) and an input/output controller hub (ICH).

40. The method of claim 28 wherein the storage is in an input/output controller hub (ICH) external to the processor.

41-60. (canceled)

61. (previous 1.116 amendment request #1) A system comprising:

a processor capable of selectively operating in a normal execution mode and, alternatively, in an isolated execution mode;

a memory coupled to the processor having an isolated memory area accessible to the processor in the isolated execution mode;

a processor executive (PE) executable on the processor to load an operating system executive (OSE) in a secure environment, the secure environment having a fused key (FK) and associated with the isolated memory area, the OSE to manage a subset of an operating system (OS);

a PE supplement residing in storage within the system, the PE supplement comprising a PE manifest that represents the PE; and

a PE handler to verify the PE using the FK and the PE supplement.

62. The system of claim 61 further comprising:

a boot-up code to load the PE handler into the isolated memory area during a process of booting up the platform.

63. The system of claim 61 wherein the secure environment includes an OSE supplement comprising an OSE manifest that represents the OSE.

64. The system of claim 61 wherein the PE handler comprises:

a PE loader to load the PE into the isolated memory area; and

a verifier to verify the PE using the PE manifest.

65. The system of claim 61 wherein the PE handler comprises:

a PE key generator to generate a PE key using the FK;

a PE identifier logger to log a PE identifier in a storage; and

a PE entrance/exit handler to handle a PE entry and a PE exit.

66. The system of claim 65 wherein the PE key generator comprises:

a PE key combiner to combine the PE identifier and the FK, the combined PE identifier and FK corresponding to the PE key.

67. The system of claim 63 wherein the PE comprises:

an OSE loader to load the OSE and the OSE supplement into the isolated memory area;

an OSE manifest verifier to verify the OSE manifest; and

an OSE verifier to verify the OSE.

68. The system of claim 61 wherein the PE comprises:

an OSE key generator to generate an OSE key;

an OSE identifier logger to log an OSE identifier in a storage; and

an OSE entrance/exit handler to handle an OSE entry and an OSE exit.

69. The system of claim 68 wherein the OSE key generator comprises:

a binding key generator to generate a binding key (BK) using a PE key; and

an OSE key combiner to combine the OSE identifier and the BK, the combined OSE identifier and BK corresponding to the OSE key.

70. The system of claim 61 wherein the OSE comprises:

a module loader to load a module into the isolated memory area;

a page manager to manage paging in the isolated memory area; and

an interface handler to handle interfacing with the OS.

71. The system of claim 70 wherein the module comprises one or more modules selected from the group consisting of an application module, an applet module, and a support module.

72. The system of claim 71 wherein the OSE further comprises:

an applet key generator to generate an applet key associated with the applet module.

73. The system of claim 72 wherein the applet key generator comprises:

an applet key combiner to combine an OSE key with an applet identifier identifying the applet module, the combined OSE key and applet identifier corresponding to the applet key.

74. The system of claim 64 wherein the boot-up code comprises:

a PE locator to locate the PE and the PE supplement, the PE locator transferring the PE and the PE supplement into the PE memory at a PE address;

a PE recorder to record the PE address in a parameter block; and

an instruction invoker to execute an isolated create instruction, the isolated create instruction loading the PE handler into the isolated memory area.

75. The system of claim 74 wherein the isolated create instruction performs an atomic sequence, the atomic sequence being non-interruptible.

76. The system of claim 75 wherein the atomic sequence includes operations comprising:

reading a thread count register in a chipset to determine if the processor is the first processor in the isolated execution mode;

configuring the processor in the isolated execution mode; and

loading the PE handler into the isolated memory area.

77. The system of claim 75 wherein the atomic sequence of operations comprises:

verifying a loaded PE handler; and

transferring control to the loaded PE handler.

78. The system of claim 76 wherein the atomic sequence of operations further comprises:

reading a configuration storage in the chipset when the processor is not the first processor in the isolated execution mode; and

configuring the processor according to the configuration storage in the chipset when the processor is not the first processor in the isolated execution mode.

79. The system of claim 78 wherein the chipset includes at least one hub selected from the group consisting of a memory controller hub (MCH) and an input/output controller hub (ICH).

80. The system of claim 68 wherein the storage is in an input/output controller hub (ICH) external to the processor.

81. (previous 1.116 amendment request #2) An apparatus comprising:

a machine accessible medium; and

instructions encoded in the machine accessible medium, wherein the instructions, when executed in a platform, cause the platform to perform operations comprising:

loading an operating system executive (OSE) into an isolated memory area of a platform, the platform having a fused key (FK) and a processor capable of selectively operating in a normal execution mode and, alternatively, in an isolated execution mode, the OSE to manage a subset of an operating system (OS) running on the platform, the isolated memory area being accessible to the processor in the isolated execution mode, the loading of the OSE initiated by a processor executive (PE) executing on the processor; and

verifying the PE using the FK and a PE supplement having a PE manifest that represents the PE, the verification to be performed by a PE handler.

82. (previous 1.116 amendment requests #1 & #2 cancel each other out) An apparatus according to claim 81, wherein the instructions implement boot-up code that performs operations comprising:

loading the PE handler into the isolated memory area during a process of booting up the platform.

83. (previous 1.116 amendment request #1) An apparatus according to claim 81, wherein the instructions implement a PE handler that performs operations comprising:

loading the PE into the isolated memory area; and

verifying the PE ~~manifest~~ using the PE manifest.

84. An apparatus according to claim 81, wherein the instructions implement a PE handler that performs operations comprising:

generating a PE key using the FK;

logging a PE identifier in a storage; and

handling a PE entry and a PE exit.

85. An apparatus according to claim 84, wherein the PE handler generates the PE key based at least in part on a combination of the PE identifier and the FK.

86. An apparatus according to claim 81, wherein the instructions cause the platform to verify the OSE after loading the OSE into the isolated memory area.

87. An apparatus according to claim 81, wherein the instructions implement the PE, and the operations performed by the PE comprise:

generating an OSE key;

logging an OSE identifier in a storage; and

handling an OSE entry and an OSE exit.

88. An apparatus according to claim 87, wherein the PE stores the OSE identifier in an input/output controller hub (ICH) external to the processor.

89. (previous 1.116 amendment request #1) An apparatus according to claim 81, wherein the instructions cause the platform to perform operations comprising:

generating a binding key (BK) using ~~the~~ a PE key; and

generating ~~the~~ an OSE key based at least in part on a combination of ~~the~~ an OSE identifier and the BK.

90. An apparatus according to claim 81, wherein the instructions implement the OSE, and the OSE manages the subset of the OS by performing operations comprising:

loading a module into the isolated memory area;

managing paging in the isolated memory area; and

interfacing with the OS.

91. An apparatus according to claim 90, wherein the module loaded by the OSE comprises one or more modules selected from the group consisting of an application module, an applet module, and a support module.

92. An apparatus according to claim 91 wherein the OSE performs further operations comprising:

generating an applet key associated with the applet module.

93. An apparatus according to claim 92, wherein the OSE generates the applet key based at least in part on a combination of an OSE key with an applet identifier identifying the applet module.

94. (previous 1.116 amendment requests #1 & #2) An apparatus according to claim 81, wherein the instructions cause the platform to perform operations comprising:

locating the PE and the PE supplement;

transferring the PE and the PE supplement into ~~the~~ PE memory at a PE address during a process of booting the platform;

recording the PE address in a parameter block; and

executing an isolated create instruction during the process of booting the platform, the isolated create instruction loading the PE handler into the isolated memory area.

95. An apparatus according to claim 94, wherein executing the isolated create instruction comprises performing an atomic sequence, the atomic sequence being non-interruptible.

96. An apparatus according to claim 95, wherein performing the atomic sequence comprises:

reading a thread count register in a chipset to determine if the processor is the first processor in the isolated execution mode;

configuring the processor in the isolated execution mode; and

loading the PE handler into the isolated memory area.

97. An apparatus according to claim 95, wherein performing the atomic sequence comprises:

verifying a loaded PE handler; and

transferring control to the loaded PE handler.

98. (previous 1.116 amendment request #1) An apparatus according to ~~claim 95~~ claim 96, wherein configuring the processor in the isolated execution mode comprises:

reading a configuration storage in the chipset when the processor is not the first processor in the isolated execution mode; and

configuring the processor according to the configuration storage in the chipset when the processor is not the first processor in the isolated execution mode.

99. (previous 1.116 amendment request #1) An apparatus according to ~~claim 95~~ claim 96, wherein the chipset includes at least one hub selected from the group consisting of a memory controller hub (MCH) and an input/output controller hub (ICH).

IX. EVIDENCE APPENDIX

Not Applicable

X. RELATED PROCEEDINGS APPENDIX

Not Applicable

XI. SUPPLEMENTAL CLAIMS APPENDIX

As indicated above, this Supplemental Claims Appendix includes the claims as they existed before Appellant submitted the two requests for amendments after final to put the claims in better form for consideration on appeal. However, it is anticipated that the requested amendments will be admitted, in which case the listing of claims in section VIII above will reflect the claims as pending on appeal, and this listing will only be of historical relevance. Accordingly, each of the pending claims listed below includes the parenthetical "(version prior to 1.116 amendments)."

1. (version prior to 1.116 amendments) An apparatus comprising:

a processor executive (PE) executable on a processor to load an operating system executive (OSE) in a secure environment, the secure environment having a fused key (FK) and associated with an isolated memory area in a platform having the processor, the OSE to manage a subset of an operating system (OS) running on the platform, the processor capable of selectively operating in a normal execution mode and, alternatively, in an isolated execution mode, the isolated memory area being accessible to the processor in the isolated execution mode;

a PE supplement comprising a PE manifest that represents the PE; and

a PE handler to verify the PE using the FK and the PE supplement.

2. (version prior to 1.116 amendments) The apparatus of claim 1 further comprising:

a boot-up code to load the PE handler into the isolated memory area during a process of booting up the platform.

3. (version prior to 1.116 amendments) The apparatus of claim 1 wherein the secure environment includes an OSE supplement comprising an OSE manifest that represents the OSE.

4. (version prior to 1.116 amendments) The apparatus of claim 1 wherein the PE handler comprises:

a PE loader to load the PE into the isolated memory area; and

a verifier to verify the PE using the PE manifest.

5. (version prior to 1.116 amendments) The apparatus of claim 1 wherein the PE handler comprises:

a PE key generator to generate a PE key using the FK;

a PE identifier logger to log a PE identifier in a storage; and

a PE entrance/exit handler to handle a PE entry and a PE exit.

6. (version prior to 1.116 amendments) The apparatus of claim 5 wherein the PE key generator comprises:

a PE key combiner to combine the PE identifier and the FK, the combined PE identifier and FK corresponding to the PE key.

7. (version prior to 1.116 amendments) The apparatus of claim 3 wherein the PE comprises:

an OSE loader to load the OSE and the OSE supplement into the isolated memory area;

an OSE manifest verifier to verify the OSE manifest; and

an OSE verifier to verify the OSE.

8. (version prior to 1.116 amendments) The apparatus of claim 1 wherein the PE comprises:

an OSE key generator to generate an OSE key;

an OSE identifier logger to log an OSE identifier in a storage; and

an OSE entrance/exit handler to handle an OSE entry and an OSE exit.

9. (version prior to 1.116 amendments) The apparatus of claim 8 wherein the OSE key generator comprises:

a binding key generator to generate a binding key (BK) using a PE key; and

an OSE key combiner to combine the OSE identifier and the BK, the combined OSE identifier and BK corresponding to the OSE key.

10. (version prior to 1.116 amendments) The apparatus of claim 1 wherein the OSE comprises:

a module loader to load a module into the isolated memory area;

a page manager to manage paging in the isolated memory area; and

an interface handler to handle interfacing with the OS.

11. (version prior to 1.116 amendments) The apparatus of claim 10 wherein the module comprises one or more modules selected from the group consisting of an application module, an applet module, and a support module.

12. (version prior to 1.116 amendments) The apparatus of claim 11 wherein the OSE further comprises:

an applet key generator to generate an applet key associated with the applet module.

13. (version prior to 1.116 amendments) The apparatus of claim 12 wherein the applet key generator comprises:

an applet key combiner to combine an OSE key with an applet identifier identifying the applet module, the combined OSE key and applet identifier corresponding to the applet key.

14. (version prior to 1.116 amendments) The apparatus of claim 4 wherein the boot-up code comprises:

a PE locator to locate the PE and the PE supplement, the PE locator transferring the PE and the PE supplement into the PE memory at a PE address;

a PE recorder to record the PE address in a parameter block; and

an instruction invoker to execute an isolated create instruction, the isolated create instruction loading the PE handler into the isolated memory area.

15. (version prior to 1.116 amendments) The apparatus of claim 14 wherein the isolated create instruction performs an atomic sequence, the atomic sequence being non-interruptible.

16. (version prior to 1.116 amendments) The apparatus of claim 15 wherein the atomic sequence includes operations comprising:

reading a thread count register in a chipset to determine if the processor is the first processor in the isolated execution mode;

configuring the processor in the isolated execution mode; and

loading the PE handler into the isolated memory area.

17. (version prior to 1.116 amendments) The apparatus of claim 15 wherein the atomic sequence of operations comprises:

verifying a loaded PE handler; and

transferring control to the loaded PE handler.

18. (version prior to 1.116 amendments) The apparatus of claim 16 wherein the atomic sequence of operations further comprises:

reading a configuration storage in the chipset when the processor is not the first processor in the isolated execution mode; and

configuring the processor according to the configuration storage in the chipset when the processor is not the first processor in the isolated execution mode.

19. (version prior to 1.116 amendments) The apparatus of claim 18 wherein the chipset includes at least one hub selected from the group consisting of a memory controller hub (MCH) and an input/output controller hub (ICH).

20. (version prior to 1.116 amendments) The apparatus of claim 8 wherein the storage is in an input/output controller hub (ICH) external to the processor.

21. (version prior to 1.116 amendments) A method comprising:

loading an operating system executive (OSE) into an isolated memory area of a platform, the platform having a fused key (FK) and a processor capable of selectively operating in a normal execution mode and, alternatively, in an isolated execution mode, the OSE to manage a subset of an operating system (OS) running on the platform, the isolated memory area being accessible to the processor in the isolated execution mode, the loading of the OSE initiated by a processor executive (PE) executing on the processor; and

verifying the PE using the FK and a PE supplement having a PE manifest that represents the PE.

22. (version prior to 1.116 amendments) The method of claim 21 further comprising:

loading the PE into the isolated memory area during a process of booting up the platform.

23. (canceled)

24. (version prior to 1.116 amendments) The method of claim 21 wherein the PE handler performs operations comprising:

loading the PE into the isolated memory area; and

verifying the PE using the PE manifest.

25. (version prior to 1.116 amendments) The method of claim 24 wherein the PE handler performs operations comprising:

generating a PE key using the FK;

logging a PE identifier in a storage; and

handling a PE entry and a PE exit.

26. (version prior to 1.116 amendments) The method of claim 25 wherein generating the PE key comprises:

combining the PE identifier and the FK, the combined PE identifier and FK corresponding to the PE key.

27. (version prior to 1.116 amendments) The method of claim 21, further comprising:

and

verifying the OSE after loading the OSE into the isolated memory area.

28. (version prior to 1.116 amendments) The method of claim 21 wherein the operations performed by the PE comprise:

generating an OSE key;

logging an OSE identifier in a storage; and

handling an OSE entry and an OSE exit.

29. (version prior to 1.116 amendments) The method of claim 28 wherein generating the OSE key comprises:

generating a binding key (BK) using the PE key; and

combining the OSE identifier and the BK, the combined OSE identifier and BK corresponding to the OSE key.

30. (version prior to 1.116 amendments) The method of claim 21 wherein the OSE manages the subset of the OS by performing operations comprising:

loading a module into the isolated memory area;

managing paging in the isolated memory area; and

interfacing with the OS.

31. (version prior to 1.116 amendments) The method of claim 29 wherein the module comprises one or more modules selected from the group consisting of an application module, an applet module, and a support module.

32. (version prior to 1.116 amendments) The method of claim 31 wherein the OSE performs further operations comprising:

generating an applet key associated with the applet module.

33. (version prior to 1.116 amendments) The method of claim 32 wherein:

the OSE combines an OSE key with an applet identifier identifying the applet module, the combined OSE key and applet identifier corresponding to the applet key.

34. (version prior to 1.116 amendments) The method of claim 21, further comprising:

locating the PE and the PE supplement;

transferring the PE and the PE supplement into the PE memory at a PE address during a process of booting the platform;

recording the PE address in a parameter block; and

executing an isolated create instruction during the process of booting the platform, the isolated create instruction loading the PE handler into the isolated memory area.

35. (version prior to 1.116 amendments) The method of claim 34 wherein executing the isolated create instruction comprises performing an atomic sequence, the atomic sequence being non-interruptible.

36. (version prior to 1.116 amendments) The method of claim 35 wherein performing the atomic sequence comprises:

reading a thread count register in a chipset to determine if the processor is the first processor in the isolated execution mode;

configuring the processor in the isolated execution mode; and

loading the PE handler into the isolated memory area.

37. (version prior to 1.116 amendments) The method of claim 35 wherein performing the atomic sequence comprises:

verifying a loaded PE handler; and

transferring control to the loaded PE handler.

38. (version prior to 1.116 amendments) The method of claim 36 wherein configuring the processor in the isolated execution mode comprises:

reading a configuration storage in the chipset when the processor is not the first processor in the isolated execution mode; and

configuring the processor according to the configuration storage in the chipset when the processor is not the first processor in the isolated execution mode.

39. (version prior to 1.116 amendments) The method of claim 38 wherein the chipset includes at least one hub selected from the group consisting of a memory controller hub (MCH) and an input/output controller hub (ICH).

40. (version prior to 1.116 amendments) The method of claim 28 wherein the storage is in an input/output controller hub (ICH) external to the processor.

41-60. (canceled)

61. (version prior to 1.116 amendments) A system comprising:

a processor capable of selectively operating in a normal execution mode and, alternatively, in an isolated execution mode;

a memory coupled to the processor having an isolated memory area accessible to the processor in the isolated execution mode;

a processor executive (PE) executable on the processor to load an operating system executive (OSE) in a secure environment, the secure environment having a fused key (FK) and associated with the isolated memory, the OSE to manage a subset of an operating system (OS);

a PE supplement residing in storage within the system, the PE supplement comprising a PE manifest that represents the PE; and

a PE handler to verify the PE using the FK and the PE supplement.

62. (version prior to 1.116 amendments) The system of claim 61 further comprising:

a boot-up code to load the PE handler into the isolated memory area during a process of booting up the platform.

63. (version prior to 1.116 amendments) The system of claim 61 wherein the secure environment includes an OSE supplement comprising an OSE manifest that represents the OSE.

64. (version prior to 1.116 amendments) The system of claim 61 wherein the PE handler comprises:

a PE loader to load the PE into the isolated memory area; and

a verifier to verify the PE using the PE manifest.

65. (version prior to 1.116 amendments) The system of claim 61 wherein the PE handler comprises:

a PE key generator to generate a PE key using the FK;

a PE identifier logger to log a PE identifier in a storage; and

a PE entrance/exit handler to handle a PE entry and a PE exit.

66. (version prior to 1.116 amendments) The system of claim 65 wherein the PE key generator comprises:

a PE key combiner to combine the PE identifier and the FK, the combined PE identifier and FK corresponding to the PE key.

67. (version prior to 1.116 amendments) The system of claim 63 wherein the PE comprises:

an OSE loader to load the OSE and the OSE supplement into the isolated memory area;

an OSE manifest verifier to verify the OSE manifest; and

an OSE verifier to verify the OSE.

68. (version prior to 1.116 amendments) The system of claim 61 wherein the PE comprises:

an OSE key generator to generate an OSE key;

an OSE identifier logger to log an OSE identifier in a storage; and

an OSE entrance/exit handler to handle an OSE entry and an OSE exit.

69. (version prior to 1.116 amendments) The system of claim 68 wherein the OSE key generator comprises:

a binding key generator to generate a binding key (BK) using a PE key; and

an OSE key combiner to combine the OSE identifier and the BK, the combined OSE identifier and BK corresponding to the OSE key.

70. (version prior to 1.116 amendments) The system of claim 61 wherein the OSE comprises:

a module loader to load a module into the isolated memory area;

a page manager to manage paging in the isolated memory area; and

an interface handler to handle interfacing with the OS.

71. (version prior to 1.116 amendments) The system of claim 70 wherein the module comprises one or more modules selected from the group consisting of an application module, an applet module, and a support module.

72. (version prior to 1.116 amendments) The system of claim 71 wherein the OSE further comprises:

an applet key generator to generate an applet key associated with the applet module.

73. (version prior to 1.116 amendments) The system of claim 72 wherein the applet key generator comprises:

an applet key combiner to combine an OSE key with an applet identifier identifying the applet module, the combined OSE key and applet identifier corresponding to the applet key.

74. (version prior to 1.116 amendments) The system of claim 64 wherein the boot-up code comprises:

a PE locator to locate the PE and the PE supplement, the PE locator transferring the PE and the PE supplement into the PE memory at a PE address;

a PE recorder to record the PE address in a parameter block; and

an instruction invoker to execute an isolated create instruction, the isolated create instruction loading the PE handler into the isolated memory area.

75. (version prior to 1.116 amendments) The system of claim 74 wherein the isolated create instruction performs an atomic sequence, the atomic sequence being non-interruptible.

76. (version prior to 1.116 amendments) The system of claim 75 wherein the atomic sequence includes operations comprising:

reading a thread count register in a chipset to determine if the processor is the first processor in the isolated execution mode;

configuring the processor in the isolated execution mode; and

loading the PE handler into the isolated memory area.

77. (version prior to 1.116 amendments) The system of claim 75 wherein the atomic sequence of operations comprises:

verifying a loaded PE handler; and

transferring control to the loaded PE handler.

78. (version prior to 1.116 amendments) The system of claim 76 wherein the atomic sequence of operations further comprises:

reading a configuration storage in the chipset when the processor is not the first processor in the isolated execution mode; and

configuring the processor according to the configuration storage in the chipset when the processor is not the first processor in the isolated execution mode.

79. (version prior to 1.116 amendments) The system of claim 78 wherein the chipset includes at least one hub selected from the group consisting of a memory controller hub (MCH) and an input/output controller hub (ICH).

80. (version prior to 1.116 amendments) The system of claim 68 wherein the storage is in an input/output controller hub (ICH) external to the processor.

81. (version prior to 1.116 amendments) An apparatus comprising:

a machine accessible medium; and

instructions encoded in the machine accessible medium, wherein the instructions, when executed in a platform, cause the platform to perform operations comprising:

loading an operating system executive (OSE) into an isolated memory area of a platform, the platform having a fused key (FK) and a processor capable of selectively operating in a normal execution mode and, alternatively, in an isolated execution mode, the OSE to manage a subset of an operating system (OS) running on the platform, the isolated memory area being accessible to the processor in the isolated execution mode, the loading of the OSE initiated by a processor executive (PE) executing on the processor; and

verifying the PE using the FK and a PE supplement having a PE manifest that represents the PE.

82. (version prior to 1.116 amendments) An apparatus according to claim 81, wherein the instructions implement boot-up code that performs operations comprising:

loading the PE handler into the isolated memory area during a process of booting up the platform.

83. (version prior to 1.116 amendments) An apparatus according to claim 81, wherein the instructions implement a PE handler that performs operations comprising:

loading the PE into the isolated memory area; and

verifying the PE manifest using the PE manifest.

84. (version prior to 1.116 amendments) An apparatus according to claim 81, wherein the instructions implement a PE handler that performs operations comprising:

generating a PE key using the FK;

logging a PE identifier in a storage; and

handling a PE entry and a PE exit.

85. (version prior to 1.116 amendments) An apparatus according to claim 84, wherein the PE handler generates the PE key based at least in part on a combination of the PE identifier and the FK.

86. (version prior to 1.116 amendments) An apparatus according to claim 81, wherein the instructions cause the platform to verify the OSE after loading the OSE into the isolated memory area.

87. (version prior to 1.116 amendments) An apparatus according to claim 81, wherein the instructions implement the PE, and the operations performed by the PE comprise:

generating an OSE key;

logging an OSE identifier in a storage; and

handling an OSE entry and an OSE exit.

88. (version prior to 1.116 amendments) An apparatus according to claim 87, wherein the PE stores the OSE identifier in an input/output controller hub (ICH) external to the processor.

89. (version prior to 1.116 amendments) An apparatus according to claim 81, wherein the instructions cause the platform to perform operations comprising:

generating a binding key (BK) using the PE key; and

generating the OSE key based at least in part on a combination of the OSE identifier and the BK.

90. (version prior to 1.116 amendments) An apparatus according to claim 81, wherein the instructions implement the OSE, and the OSE manages the subset of the OS by performing operations comprising:

loading a module into the isolated memory area;

managing paging in the isolated memory area; and

interfacing with the OS.

91. (version prior to 1.116 amendments) An apparatus according to claim 90, wherein the module loaded by the OSE comprises one or more modules selected from the group consisting of an application module, an applet module, and a support module.

92. (version prior to 1.116 amendments) An apparatus according to claim 91 wherein the OSE performs further operations comprising:

generating an applet key associated with the applet module.

93. (version prior to 1.116 amendments) An apparatus according to claim 92, wherein the OSE generates the applet key based at least in part on a combination of an OSE key with an applet identifier identifying the applet module.

94. (version prior to 1.116 amendments) An apparatus according to claim 81, wherein the instructions cause the platform to perform operations comprising:

locating the PE and the PE supplement;

transferring the PE and the PE supplement into the PE memory at a PE address during a process of booting the platform;

recording the PE address in a parameter block; and

executing an isolated create instruction during the process of booting the platform, the isolated create instruction loading the PE handler into the isolated memory area.

95. (version prior to 1.116 amendments) An apparatus according to claim 94, wherein executing the isolated create instruction comprises performing an atomic sequence, the atomic sequence being non-interruptible.

96. (version prior to 1.116 amendments) An apparatus according to claim 95, wherein performing the atomic sequence comprises:

reading a thread count register in a chipset to determine if the processor is the first processor in the isolated execution mode;

configuring the processor in the isolated execution mode; and

loading the PE handler into the isolated memory area.

97. (version prior to 1.116 amendments) An apparatus according to claim 95, wherein performing the atomic sequence comprises:

verifying a loaded PE handler; and

transferring control to the loaded PE handler.

98. (version prior to 1.116 amendments) An apparatus according to claim 95, wherein configuring the processor in the isolated execution mode comprises:

reading a configuration storage in the chipset when the processor is not the first processor in the isolated execution mode; and

configuring the processor according to the configuration storage in the chipset when the processor is not the first processor in the isolated execution mode.

99. (version prior to 1.116 amendments) An apparatus according to claim 95, wherein the chipset includes at least one hub selected from the group consisting of a memory controller hub (MCH) and an input/output controller hub (ICH).